

Laboratory 14

Mobilité urbaine et villes intelligentes : simulation des horaires de bus

Module: Sujets avancés et études de cas

Durée: 135 Minutes

Contents

Objective	2
1 Scénario de simulation et fichiers	3
2 Configuration et exécution initiale	4
3 Activité 1 – Référence statique vs référence intelligente	5
4 Activité 2 – Balayage des paramètres (Expérience A)	6
5 Activité 3 – Test de résistance et vérification des capitaux propres (Expérience B)	7
6 Devoirs – Réglage et essai de la route R12	9

Objective

Dans ce laboratoire, vous utiliserez un simulateur Python pour modéliser une ligne de bus (**R12**) sur une journée complète. Vous serez:

- Exécutez le simulateur avec un programme statique (intervalle fixe de 20 minutes),,
- Exécutez-le avec une politique d'intervalle intelligent (10 min en heures de pointe, 20 min en heures creuses),
- comparer les performances à l'aide d'indicateurs clés (temps d'attente, passagers servis),
- effectuer un petit balayage de paramètres,
- et effectuer un test de résistance et une vérification des capitaux propres au niveau d'arrêt.

L'accent est mis sur le respect des étapes pratiques et l'interprétation des résultats.

1 Scénario de simulation et fichiers

Scénario de la route R12

- **Itinéraire:** Ligne de bus unique R12 reliant le centre-ville à la banlieue.
- **horizon temporel:** 24 heures (0-1440 minutes).
- **Arrêts:** 20 arrêts, indexés de 0 à 19.
- **Passagers:** Les passagers se présentent à chaque arrêt de manière aléatoire. Ils montent à bord du premier bus qui arrive après leur heure d'arrivée, sous réserve de places disponibles.
- **Comparaison:**
 - **Horaire statique:** Des bus toutes les 20 minutes, toute la journée.
 - **Programme intelligent:** bus toutes les 10 minutes aux heures de pointe (08:00-18:00), toutes les 20 minutes en dehors des heures de pointe.

Bloc-notes et principales variables

Vous travaillerez dans le cahier :

- `Urban_Mobilities_and_Smart_Cities_Lab_Session.ipynb`

Variables de configuration importantes en haut du bloc-notes :

```
simulation_time      = 1440  # 24 hours
initial_bus_interval = 20    # min, baseline headway
bus_capacity         = 80
stop_positions       = np.arange(0, 20, 1)
passenger_arrival_rate = 5    # min between passenger arrivals

np.random.seed(42)      # do NOT change for the main activities
```

Le simulateur génère un tableau comportant (entre autres) les colonnes suivantes:

- `stop` – Indice d'arrêt (0–19),
- `arrival_time` – lorsque le passager est arrivé à l'arrêt,
- `bus_time` – heure de départ du bus à bord duquel le passager est monté,
- `wait_time` – temps d'attente en minutes,
- `type` – "Static" ou "Smart".

À la fin du cahier, vous verrez :

- `combined_df` – tous les passagers, dans les deux modes de transport,
- `summary_df` – indicateurs clés de performance par mode,
- `hourly_avg_reset` – temps d'attente moyens horaires.

Réflexion et perspicacité

Lorsque vous modifiez les paramètres d'une expérience, ne les modifiez qu'un seul à la fois et conservez la graine aléatoire à 42 afin que les résultats soient comparables.

2 Configuration et exécution initiale

Tâche 1: Ouvrir le bloc-notes et vérifier les paramètres

Objectif: Assurez-vous que tout le monde parte de la même configuration de base.

Mesures:

1. Ouvrez le cahier `Urban_Mobilities_and_Smart_Cities_Lab_Session.ipynb` dans Google Colab ou Jupyter.
2. Trouvez la première cellule de code avec la configuration (cherchez `simulation_time`, `initial_bus_interval`, etc.).
3. Veuillez confirmer que les valeurs sont:
 - `simulation_time = 1440`
 - `initial_bus_interval = 20`
 - `bus_capacity = 80`
 - `stop_positions = np.arange(0, 20, 1)`
 - `passenger_arrival_rate = 5`
 - `np.random.seed(42)`
4. Si l'un de ces éléments diffère, modifiez-le maintenant.

Point de contrôle: Vous devriez pouvoir expliquer en une phrase ce que chacun de ces paramètres contrôle.

Tâche 2: Exécuter toutes les cellules une seule fois

Objectif: Assurez-vous que la simulation complète (statique + intelligente) s'exécute sans erreur.

Étapes :

1. Dans le menu du notebook, choisissez **Runtime** → **Restart and run all** (ou l'équivalent dans votre environnement).
2. Attendez que toutes les cellules soient exécutées. Cela peut prendre un court moment selon votre machine.
3. Faites défiler jusqu'en bas du notebook et confirmez que vous voyez :
 - des statistiques récapitulatives imprimées (un tableau nommé quelque chose comme `summary_df`),
 - des moyennes horaires imprimées,
 - plusieurs graphiques (boxplot, courbe horaire, carte de chaleur),
 - des fichiers CSV créés : `r12_results.csv`, `r12_summary.csv`, `r12_hourly_avg.csv`.

Réflexion et perspicacité

Si vous voyez des erreurs, ne *bricolez pas* des lignes au hasard. Commencez par *redémarrer* le noyau, vérifiez la cellule des paramètres, puis relancez tout.

3 Activité 1 – Référence statique vs référence intelligente

Dans cette activité, vous utilisez les paramètres par défaut pour comprendre le comportement global des deux horaires et extraire des KPI de base.

Tâche 3: Lire et enregistrer les indicateurs clés de performance de référence

Objectif : Extraire les indicateurs clés de performance pour les horaires Static et Smart.

Étapes :

1. Repérez le tableau imprimé `summary_df`. Il doit contenir une ligne pour **Static** et une pour **Smart**.
2. Dans vos notes de labo (ou ci-dessous), notez pour chaque mode :
 - le temps d'attente moyen,
 - le temps d'attente maximal,
 - le nombre de passagers servis.

Modèle :

Mode	Attente moyenne (min)	Attente max (min)	Passagers servis
Static	<i>(copier depuis summary_df)</i>	<i>(copier)</i>	<i>(copier)</i>
Smart	<i>(copier depuis summary_df)</i>	<i>(copier)</i>	<i>(copier)</i>

Petit calcul :

- Calculez (Attente moyenne Smart) moins (Attente moyenne Static).
- Calculez (Passagers Smart) moins (Passagers Static).

Réflexion et perspicacité

Si Smart a une attente moyenne plus faible mais ne sert que quelques passagers de plus, il se peut qu'il ne fasse que lisser les pics. S'il sert beaucoup plus de passagers et réduit les attentes, la politique est clairement bénéfique.

Tâche 4 : Examiner les intrigues principales

Objectif : Comparer visuellement les performances de Static et Smart.

Étapes :

1. Trouvez le **boxplot** comparant les temps d'attente **Static** vs **Smart**.
 - Notez quel mode a la médiane la plus basse.
 - Notez quel mode a la moustache supérieure la plus longue (pires attentes).
2. Trouvez le graphique du **temps d'attente moyen par heure**.
 - Identifiez les heures où les temps d'attente sont les plus élevés.
 - Comparez les courbes Static et Smart pendant les heures de pointe (matin et soir).
3. Prenez une capture d'écran ou sauvegardez ces figures (vous en aurez besoin pour votre rapport de devoir).

Questions auxquelles répondre en une ou deux phrases :

- À quels moments de la journée la politique Smart réduit-elle clairement le temps d'attente ?
- Y a-t-il un moment où Static obtient un résultat similaire ou meilleur ?

4 Activité 2 – Balayage des paramètres (Expérience A)

Vous explorez maintenant la sensibilité du système à *un* paramètre. Vous ne modifierez qu'une seule valeur, relancerez le notebook, puis comparerez les KPI.

Tâche 5 : Choisir et modifier un paramètre

Objectif : Réaliser une expérience contrôlée avec une seule modification.

Étape 0 – Choisissez UN paramètre à modifier :

- **Headway :** Modifiez `initial_bus_interval` (p. ex. de 20 à 18 ou 22).
- **Capacité :** Modifiez `bus_capacity` (p. ex. de 80 à 60 ou 100).
- **Demande :** Modifiez `passenger_arrival_rate` (p. ex. de 5 à 3 minutes entre arrivées).

Étapes :

1. Dans la cellule de configuration, modifiez uniquement la valeur du paramètre choisi. Exemple (ajustement du headway) :

```
initial_bus_interval = 18 # instead of 20
```

2. Laissez tous les autres paramètres, en particulier `np.random.seed(42)`, inchangés.
3. Utilisez **Restart and run all**. Attendez que toutes les cellules se terminent.
4. À nouveau, trouvez le tableau `summary_df` et notez les nouvelles valeurs de KPI pour Static et Smart.

Optionnel : scénarios multiples

Si le temps le permet, répétez la procédure ci-dessus pour 2–3 valeurs différentes du même paramètre (p. ex. intervalles 16, 18, 20, 22). Notez les valeurs dans un tableau comme :

Scénario	Valeur du paramètre	Mode	Attente moyenne	Passagers
A1	20	Static	...	...
		Smart	...	...
A2	18	Static	...	...
		Smart	...	...

Réflexion et perspicacité

Assurez-vous de toujours savoir *quelle* exécution a produit *quels* nombres. Étiquetez clairement vos scénarios (A1, A2, ...).

5 Activité 3 – Test de résistance et vérification des capitaux propres (Expérience B)

Ici, vous poussez le système vers une situation plus extrême et vérifiez si la politique Smart reste équitable, en particulier pour les arrêts les plus défavorisés.

Tâche 6 : Définir un scénario de stress

Objectif: Simuler une condition “difficile” pour le réseau.

Choisissez UN scénario de stress (exemples) :

- **Demande plus élevée :**

```
passenger_arrival_rate = 3    # more passengers per hour
```

- **Capacité plus faible :**

```
bus_capacity = 50              # smaller bus
```

- **Fenêtre de pointe décalée :** Dans le code qui construit les headways Smart, modifiez la condition. Par exemple, de :

```
480 <= t <= 1080    # 08:00--18:00
```

à :

```
540 <= t <= 1110    # 09:00--18:30
```

Étapes :

1. Appliquez exactement une modification de stress dans la configuration ou dans le code de l'horaire Smart.
2. Redémarrez le noyau et relancez toutes les cellules.
3. Vérifiez que le notebook se termine correctement et produit de nouveaux graphiques et tableaux récapitulatifs.

Tâche 7 : Analyse des capitaux propres au niveau d'arrêt pour SMART

Objectif : Identifier quels arrêts subissent les attentes les plus longues avec la politique Smart.

Étapes :

1. Créez un data frame contenant uniquement les passagers Smart. Dans une nouvelle cellule, exécutez :

```
smart_df = combined_df[combined_df["type"] == "Smart"]
```

2. Calculez, pour chaque arrêt :
 - le temps d'attente moyen,
 - le 95e percentile du temps d'attente,
 - le nombre de passagers.

Exemple de code (adaptez les noms de colonnes si nécessaire) :

```
def p95(x):
    return x.quantile(0.95)

stop_stats = smart_df.groupby("stop")["wait_time"].agg(
    mean_wait="mean",
    p95_wait=p95,
    n_passengers="size"
)

print(stop_stats)
```

3. À partir de `stop_stats`, notez :
 - l'arrêt avec le `mean_wait` le plus élevé,
 - l'arrêt avec le `p95_wait` le plus élevé.
4. Regardez la **carte de chaleur des temps d'attente** (Arrêt vs Heure) produite par le notebook.
 - Identifiez quels arrêts et quelles heures sont les plus sombres (pires attentes).

À rédiger (2–4 phrases) :

- Quels arrêts sont les plus défavorisés avec Smart dans le cas de base ?
- Comment cela change-t-il dans votre scénario de stress ?
- Smart améliore-t-il la situation des pires arrêts, ou améliore-t-il surtout la moyenne ?

Réflexion et perspicacité

Un horaire peut sembler “bon” en moyenne tout en restant injuste pour quelques arrêts. Votre rôle ici est de le repérer.

6 Devoirs – Réglage et essai de la route R12

Pour le devoir, vous allez concevoir *une* petite amélioration ou variation et la tester avec le même simulateur. Cela deviendra un court rapport.

Tâche 8 : Choisissez une piste et exécutez votre scénario

Objectif : Mettre en œuvre une refonte clairement définie du système.

Choisissez UN morceau :

- **Piste A – Réglage Headway:** Conservez la logique intelligente, mais changez-la. `initial_bus_interval` (par exemple de 20 à 18 ans).
- **Piste B – Sensibilité de la capacité :** Conservez le même horaire, mais changez-le. `bus_capacity` (par exemple 60 ou 100).
- **Piste C – Décalage de la fenêtre de pointe:** Conservez les mêmes intervalles, mais décalez la fenêtre de pointe Smart (par exemple, de 08:00-18:00 à 09:00-18:30).

Mesures:

1. Redémarrez le noyau du notebook.
2. Appliquez la modification choisie (seulement *un*).
3. Exécutez toutes les cellules afin que les scénarios statique et intelligent soient simulés pour votre nouveau scénario.
4. Réservez:
 - le nouveau `summary_df` (capture d'écran ou valeurs),
 - graphique du temps d'attente moyen horaire (statique vs intelligent),
 - le diagramme en boîte comparant Statique et Intelligent.
5. Comparez votre scénario remanié au scénario de référence initial (résultats des tâches 3 et 4).

Votre rapport écrit doit au moins inclure les éléments suivants:

- Une brève description de votre modification (quel paramètre, ancienne valeur vs nouvelle valeur).
- Un tableau comparant les principaux indicateurs de performance (statiques vs intelligents, scénario de référence vs votre scénario).
- Un ou deux graphiques clés (moyennes horaires, diagramme en boîte).
- Une brève analyse des principaux changements et des compromis que vous observez (meilleurs poids ou plus de bus / capacité / etc.).

Réflexion et perspicacité

Soyez explicite: “J’ai changé X de A à B. Par conséquent, le temps d’attente moyen a augmenté de Y minutes et le nombre de passagers servis a varié de Z. Est-ce bon/mauvais car ...”